(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－24958

(43)公開日　平成11年(1999)１月29日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　11/28	３１０	Ｇ０６Ｆ　11/28	３１０Ａ

審査請求　有　　請求項の数10　ＦＤ　（全 23 頁）

(21)出願番号　特願平9－189322

(22)出願日　平成９年(1997)６月30日

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　米澤　直道
東京都港区芝五丁目７番１号　日本電気株式会社内

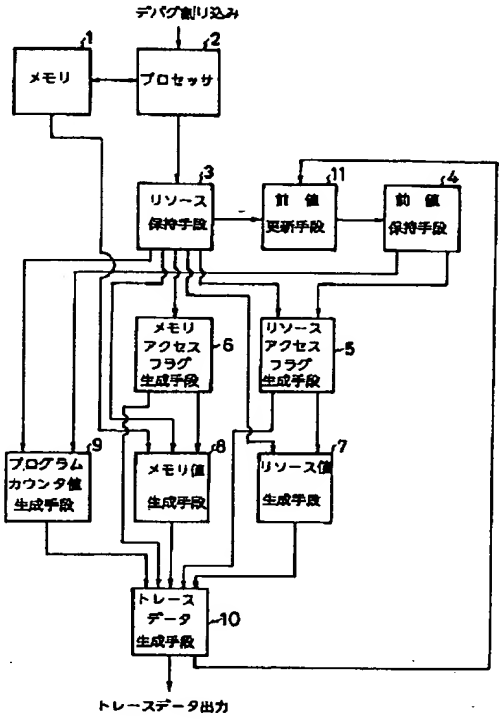
(74)代理人　弁理士　河原　純一

(54)【発明の名称】　トレースデータ削減方式

(57)【要約】

【課題】　情報処理装置の各リソースの持つ特性に注目したトレースデータの生成を行うことにより、トレースデータの情報量の削減を図る。

【解決手段】　リソースアクセスフラグ生成手段５はリソース保持手段３のリソースの値と前値保持手段４のリソースの値とを比較し、アクセスされたリソースと１対１に対応したビットパターンを生成する。リソース値生成手段７はビットパターンに基づいてリソース保持手段３からアクセスされたリソースの値を出力する。トレースデータ生成手段１０はビットパターンとリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力し、前値更新手段１１はこの指示によりリソース保持手段３のリソースの値で前値保持手段４のリソースの値を更新する。



1

【特許請求の範囲】

【請求項１】　１命令毎にデバグ割り込みを発生させることが可能であり、かつデバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段を有する情報処理装置において、

１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段と、

前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段と、

このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段と、

前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段と、

このトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段とを有することを特徴とするトレースデータ削減方式。

【請求項２】　命令語の中に、アクセスするメモリのアドレスを指定するアドレス指定フィールドを１つもしくは複数有する情報処理装置において、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段と、このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段とを有するとともに、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力する前記トレースデータ生成手段を有する請求項１記載のトレースデータ削減方式。

【請求項３】　前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとととともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成する前記トレースデータ生成手段を有する請求項１または請求項２記載のトレースデータ削減方式。

【請求項４】　前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウ

ンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成する前記トレースデータ生成手段を有する請求項１または請求項２記載のトレースデータ削減方式。

【請求項５】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項６】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項７】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了

指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項８】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項９】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ

生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【請求項１０】　コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はトレースデータ削減方式に関し、特に情報処理装置の性能を解析するために実際の情報処理装置上でユーザプログラムがどのようなシーケンスで実行されるのかを表す実行履歴であるトレースデータを削減して記録するトレースデータ削減方式に関する。

【０００２】

【従来の技術】従来、性能評価のために命令の実行履歴であるトレースデータを採取する場合、１命令毎にデバグ割り込みを発生させ、このデバグ割り込みによりスタックフレーム上などに格納された情報処理装置の持つプログラムカウンタ，命令語，レジスタなどのリソースの値を、そのまま外部記憶装置に出力していた。そのため、一般的な情報処理装置が１つの命令でアクセスできるリソースの数が１から数個程度であるのに対し、外部記憶装置には更新されていないリソースの値を含むすべてのリソースの値が記録されることになり、効率の悪いデータ列となっていた。

【０００３】そこで、１命令毎に変化したリソースの値だけを記録することにより、外部記憶装置に記憶されるデータの効率を改善することが可能であり、その従来技術として、特開平７－１９１８７５号公報（以下、先行技術文献という）に開示されたデバッガーがある。この従来のデバッガーには、プログラムを１ステップずつ実行するとともに、１命令実行する毎に情報処理装置の状態を出力するトレース実行手段と、トレース実行手段の出力する情報処理装置の状態を保持する実行状態保持手段と、トレース実行手段の出力する情報処理装置の状態と実行状態保持手段の情報とから変更履歴を作成する実行出力抽出手段と、この変更履歴を記憶する変更履歴保持手段とが含まれている。ただし、上記先行技術文献には、変更履歴の構成や構造に関する記述はない。また、先行技術文献第３頁右段の２５行目から２９行目に示される実施例の記述から、変更のあったメモリやレジスタの種類と、レジスタ番号やメモリのアドレスおよび実際

の値からなる履歴情報であると考えられる。加えて、上記先行技術文献において、実行出力抽出手段は、変更のあった情報を出力する以外、特段の機能を有していないことから、これら履歴情報は、履歴情報間の相互関係などを意識した構造になっていないと考えられる。また、変更された情報列のみを変更履歴保持手段に記憶した場合、１命令分の区切りを認識することが不可能となってしまうため、必然的に１命令分を識別するための区切り情報が必要になると考えられる。

【０００４】以上から、先行技術文献の変更履歴保持手段に記録される１つのリソースに関する履歴情報は、図１２に示すような形式になると考えられる。

【０００５】図１２は、１つの履歴情報に関するデータ形式の例であり、ここでは、これを単位情報と呼ぶ。この単位情報には、そこに保持されている情報の種類、たとえばプログラムカウンタ，レジスタ，メモリといったリソースを示す識別情報２１と、各種類毎の位置、たとえばレジスタの場合はレジスタ番号、メモリの場合はメモリのアドレスを示す位置情報２２と、情報の内容そのものを示す内容２３とが含まれる。

【０００６】図１３は、ある命令でレジスタの１番が１２３４の内容でアクセスされた場合のレジスタ１に関する単位情報の例を示す。図１３において、識別情報２１にはレジスタを示す識別情報Ｒが登録される。また、位置情報２２にはレジスタ番号の１に対応する情報が登録される。同様に、内容２３には、アクセスされたレジスタの１番の内容である１２３４が登録されることになる。

【０００７】上記先行技術文献では、１命令毎に変化のあったリソースの単位情報を実行履歴として変更履歴保持手段に残すことができるため、この実行履歴を外部記憶装置などに出力することにより、プログラムの実行履歴であるトレースデータを作成できる。

【０００８】そこで、上記先行技術文献のデバッガーを使用して、トレースデータを採取した場合、どのようなトレースデータが採取されるかについて説明する。

【０００９】まず、図２はトレースデータを採取する情報処理装置の条件、図１４はリソースと単位情報の中の識別情報との対応表を示す。

【００１０】図５は、トレースデータを採取する５つの命令の実行イメージを１命令毎に別々の表で示している。図５の表の見方は、リソースの列がアクセスされるリソースの種類、位置の列が各リソースの位置情報、フィールド位置が命令語内のアドレス指定フィールドの位置、アクセス内容がアクセスされた各リソースの値を示す１０進の値である。

【００１１】図５において、位置の欄が”－”となっているものは、その欄に対応するリソースが１つしかなく、位置情報を必要としないものである。フィールド位置の欄が”－”となっているものは、その欄に対応しているリソースがメモリではなく、アドレス指定フィールドを必要としないものである。

【００１２】たとえば、図５の１命令目の表は、命令語が１０、プログラムカウンタが１００２、レジスタ２およびレジスタ８がそれぞれ４４４および６６６の値でアクセスされ、命令語のアドレス指定フィールドの１番で示された１２０番地のアドレスのメモリが８８８の値でアクセスされたことを示している。

【００１３】図１５は、図２および図１４で示した条件により、図５で示した５つの命令を順に実行した場合にトレースデータとして採取される実行履歴を示した表である。

【００１４】図１５において、トレースデータと記述された列の、識別情報，位置情報および内容がトレースデータとして出力される値である。まず、図５に示した１命令目が実行されると、はじめに命令の区切りを示すための識別情報である”Ｓ”を持ったトレースデータが出力される。次に、１命令目ではプログラムカウンタおよび命令語が内容１００２および１０でアクセスされていることから、識別情報および内容が”Ｐ”および１００２ならびに”Ｉ”および１０となるトレースデータが出力される。また、１命令目ではレジスタ２，８およびメモリの１２０番地がそれぞれ内容４４４，６６６および８８８でアクセスされていることから、それぞれの識別情報，位置情報および内容を”Ｒ”，２および４４４と、”Ｒ”，８および６６６と、”Ｍ”，１２０および８８８としたトレースデータが出力され、これが図１５の１命令目の行に記述されたトレースデータに対応する。

【００１５】次に、図１５の２命令目が実行されると、まず、１命令目と同様に、命令の区切りを示すために区切り情報”Ｓ”を持った識別情報が出力される。次に、２命令目では、プログラムカウンタおよび命令語が１００６および４５６の内容でアクセスされていることから、識別情報および内容を”Ｐ”および１００６ならびに”Ｉ”および４５６としたトレースデータが出力される。また、２命令目では、レジスタ１５およびメモリ１の４８０番地が１３５および１０１０の内容でアクセスされていることから、それぞれの識別情報，位置情報および内容を”Ｒ”，１５および１３５ならびに”Ｍ”，４８０および１０１０としたトレースデータが出力され、これが図１５の２命令目の行にかかれたトレースデータに対応する。

【００１６】同様に、図５の３，４および５命令目を実行した場合に後で採取されるトレースデータを示したのが、図１５の３，４および５命令目の行である。

【００１７】以上から、図１５のトレースデータの列が、図５の１命令目から５命令目までを順番に実行した場合のトレースデータに対応することになる。

【００１８】ここで、この従来例におけるトレースデー

タの情報量について検討する。まず、トレースデータ上の各情報の情報量を、図７に示す。図７において、識別情報のビット数は、図１４の表から少なくとも５種類必要であることから３ビットとしている。命令語長は、図５に示されている命令語長が対応する。

【００１９】図７のトレースデータ上の情報量の条件により、従来例で採取したトレースデータの情報量を示したのが、図１５中の情報量の欄であり、各欄の値は対応する単位情報毎のビット数を示す。また、合計の行に、５命令分すべてのトレースデータの情報量を示す。したがって、この従来例では、図２および図１４に示した条件下において図５で示した５つの命令列を実行した場合、採取されるトレースデータのビット数は６２７ビットとなる。

【００２０】以上のように、上記先行技術文献のデバッガーにおいても、命令毎に１命令前との変更部分をチェックし、変更された情報のみを記憶していくことで、すべてのリソースの値を保持することなく各命令毎の実行履歴であるトレースデータを採取することが可能である。

【００２１】しかしながら、情報処理装置の持つ情報にはいくつかの特性がある。たとえば、通常、情報処理装置では、１命令毎にプログラムカウンタ，命令語，レジスタなどの複数のリソースがアクセスされる。したがって、従来例のように更新されたリソースを識別するため、１つのリソース毎に専用の識別情報を持たせると、トレースデータ上では、アクセスされた情報の数に比例した識別情報が必要になる。加えて、トレースデータへの出力が必要となるレジスタなどのすべての数は、それぞれの情報処理装置のアーキテクチャや性能解析を行いたい内容にも依存するが、数十程度の場合が一般であり、その場合、リソース毎に識別情報を与えるより、各リソースと１対１で対応したビットパターンで表現した方が、同時に複数のリソースに対するアクセスの有無を表現できることから、トレースデータ内に登録されるリソースの数に比例して増える識別情報が不要な分、全体としての情報量を減らすことが可能である。

【００２２】また、メモリのアクセスについては、一般の情報処理装置の場合、命令語の値と命令実行時のレジスタの値とが既知であれば、その命令によってアクセスされるメモリのアドレスを導き出すことが可能であり、上記先行技術文献のように、メモリの位置を実行履歴として持つ必要はなく、無駄な情報である。

【００２３】加えて、プログラムカウンタは、通常、１命令毎に更新されるとともに、多くの場合、実行された命令の命令語長でインクリメントされるため、１命令毎にプログラムカウンタ全体の値を実行履歴として記録することには無駄が多い。

【００２４】以上の点から、上記先行技術文献のように、１命令前のリソースの値と現在のリソースの値との差分から単純に実行履歴を作成する方法を流用することでも、すべてのリソースの値を保存する場合に比べ、データ量を削減したトレースデータを作成することは可能である。しかし、情報処理装置の特性を考えた場合、無駄な情報が多く含まれる情報となってしまい、数億命令単位の実ジョブのトレースデータを採取するためには、上記先行技術文献の構成を利用したトレースデータの採取方法は機能的に不十分であった。

【００２５】

【発明が解決しようとする課題】第１の問題点は、上記先行技術文献に記載されたデバッガーでは、情報処理装置の状態を示す情報の持つ特性を意識せず、単純に１命令前のリソースの値と現在のリソースの値との変更部分をその他の情報に頼ることなく単独の情報で再生できるように、リソースの値とその情報の位置やリソースの種類などを識別する情報を履歴情報として記録している点にある。その理由は、通常の情報処理装置では、１命令毎にプログラムカウンタ，命令語，レジスタなどの複数のリソースがアクセスされるため、１つのリソース毎に識別情報を付与すると、トレースデータ上ではアクセスされたリソースの数に比例して、リソースを識別する情報も増えるからである。

【００２６】また、プログラムカウンタは通常１命令毎に実行した命令語長分でインクリメントされるだけであり、プログラムカウンタが変更されても、多くの場合命令語長がわかれば、１命令前のプログラムカウンタの値に命令語長を加えることにより、新しいプログラムカウンタの値を求めることが可能であることから、プログラムカウンタの値の全てを記憶しておく必要はない。

【００２７】加えて、メモリのアクセスするアドレスは、実行された命令語の値と、その命令実行時のレジスタの値とが既知であれば、それらの値から計算することが可能であり、変更されたメモリの位置であるアドレスを記憶しておくことは無駄である。

【００２８】本発明の目的は、ユーザプログラムの性能評価に使われる命令の実行履歴であるトレースデータの採取において、１命令前のプログラムカウンタやレジスタなどのリソースの値と現在のリソースの値との差分によってトレースデータを生成する以外に、情報処理装置の各リソースの持つ特性に注目したトレースデータの生成を行うことにより、トレースデータの情報量を削減するようにしたトレースデータ削減方式を提供することにある。

【００２９】また、本発明の他の目的は、１つの命令で多くのリソースがアクセスされ、それらリソースの値がトレースデータとして出力された場合でも、１命令分のトレースデータの中にどのリソースに対応する情報が登録されているかの識別が、１つのビットパターンのみで実現できるようにし、トレースデータに出力されるリソースの数に比例して増加するリソースを識別するための

情報を排除できるようにしたトレースデータ削減方式を提供することにある。

【００３０】さらに、本発明の別の目的は、メモリの内容が更新された場合、命令語のアドレス指定フィールドに対応したビットの値と、アクセスされたメモリの内容だけがトレースデータに出力されるようにし、メモリの位置を示すアドレスがトレースデータ上に出力されないようにしたトレースデータ削減方式を提供することにある。

【００３１】さらにまた、本発明のさらに別の目的は、プログラムカウンタに関しては、多くの場合、トレースデータ上に出力されるプログラムカウンタに対応した情報は、トレースデータの中にプログラムカウンタ全体の値があるのか、１命令前の値との差分のみがあるのかを示すビットの情報および命令語長、または命令語長の情報だけとなるようにし、１命令毎に常にプログラムカウンタの値がトレースデータに出力される場合に比べ、トレースデータの情報量を大幅に減らすことができるようにしたトレースデータ削減方式を提供することにある。

【００３２】

【課題を解決するための手段】本発明のトレースデータ削減方式は、１命令毎にデバグ割り込みを発生させることが可能であり、かつデバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段を有する情報処理装置において、１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段と、前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段と、このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段と、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段と、このトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段とを有する。これにより、たとえば、１つの命令で多くのリソースがアクセスされ、それらリソースの値がトレースデータとして出力された場合でも、１命令分のトレースデータの中にどのリソースに対応する情報が登録されているかの識別が、１つのビットパターンのみで実現できるため、トレースデータに出力されるリソースの数に比例して増加するリソースを識別するための情報を排除できる。

【００３３】また、本発明のトレースデータ削減方式は、請求項１記載のトレースデータ削減方式において、命令語の中に、アクセスするメモリのアドレスを指定するアドレス指定フィールドを１つもしくは複数有する情報処理装置において、命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段と、このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段とを有するとともに、事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力する前記トレースデータ生成手段を有する。これにより、メモリの内容が更新された場合、命令語のアドレス指定フィールドに対応したビットの値と、アクセスされたメモリの値だけがトレースデータに出力されるため、メモリの位置を示すアドレスがトレースデータ上に出力されない。

【００３４】さらに、本発明のトレースデータ削減方式は、請求項１または請求項２記載のトレースデータ削減方式において、前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の

出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成する前記トレースデータ生成手段を有する。これにより、プログラムカウンタに関しては、多くの場合、トレースデータ上に出力されるプログラムカウンタに対応した情報は、トレースデータの中にプログラムカウンタ全体の値があるのか、１命令前のプログラムカウンタの値との差分のみがあるのかを示すビットの情報および命令語長、または命令語長の情報だけとなるため、１命令毎に常にプログラムカウンタの値がトレースデータに出力される場合に比べ、トレースデータの情報量を大幅に減らすことができる。

【００３５】さらにまた、本発明のトレースデータ削減方式は、請求項１または請求項２記載のトレースデータ削減方式において、前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段を有し、事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成する前記トレースデータ生成手段を有する。

【００３６】一方、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースアクセスフラグ生成手段の出力するビットパターンと前記リソース値生成手段の出力するリソースの値とから、事前に決められた規則に従い、どのリソースがアクセスされたかを示すリソースビットパターンと更新されたリソースの値とで構成されるトレースデータを出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００３７】また、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，事前に決められた規則に従い前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンとを組み合わせたリソースビットパターンと、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値とをトレースデータとして出力すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指

示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００３８】さらに、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとととともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００３９】さらにまた、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットの値を、トレースデータ内にプログラムカウンタ全体の値を保持するか、または１命令前のプログラムカウンタの値との差分のみを保持するかのフラグを定義するとととともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との差分のみを出力するかを判断し、リソースビットパターン上のプログラムカウンタに対応したビットの情報と、プログラムカウンタ全体の値または差分を出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するリソースビットパターン上のプログラムカウンタに対応したビットの値とからリソースビットパターンを生成するとともに、前記リソース値生成手段の出力するリソースの値と前記メモリ値生成手段の出力するメモリの値と前記プログラムカウンタ値生成手段の出力するプログラムカウンタ全体の値または差分とからトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００４０】また、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース保

持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００４１】さらに、本発明の記録媒体は、コンピュータを、デバグ割り込み発生時のプログラムカウンタ，命令語，レジスタなどのリソースの値を保持するリソース

保持手段，１命令前のデバグ割り込み発生時のリソースの値を保持する前値保持手段，前記リソース保持手段の保持するリソースの値と前記前値保持手段の保持するリソースの値とを比較し、どのリソースがアクセスされたかを示す、リソースと１対１に対応したビットパターンを生成するリソースアクセスフラグ生成手段，このリソースアクセスフラグ生成手段の出力するビットパターンに基づいて前記リソース保持手段からアクセスされたリソースの値を出力するリソース値生成手段，命令語の中のどのアドレス指定フィールドで指定されたアドレスにアクセスがあったかを示す、アドレス指定フィールドと１対１に対応したビットパターンを生成するメモリアクセスフラグ生成手段，このメモリアクセスフラグ生成手段の出力するビットパターンと前記リソース保持手段の保持するリソースの値とからアドレス指定フィールドで指定されたメモリのアドレスを求め、該アドレスによりアクセスされたメモリの値を取り出して出力するメモリ値生成手段，前記リソースビットパターンのうちのプログラムカウンタに対応するビットを１ビットまたは複数ビットのプログラムカウンタ更新パターンに置き換え、このプログラムカウンタ更新パターンの組み合わせにより、プログラムカウンタの値が１命令前のプログラムカウンタに対し、どの程度の変位を持った値であるか、または後続するトレースデータ内にプログラムカウンタ全体の値があるかを指示できるようにするとともに、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値とを比較し、前記リソース保持手段の保持するプログラムカウンタの値をトレースデータに出力するか、前記リソース保持手段の保持するプログラムカウンタの値と前記前値保持手段の保持するプログラムカウンタの値との変位に対応したプログラムカウンタ更新パターンのみを出力するかを判断し、プログラムカウンタ全体の値を出力する場合はトレースデータ内にプログラムカウンタ全体の値があることを示すプログラムカウンタ更新パターンとプログラムカウンタ全体の値とを出力し、プログラムカウンタの値の変位のみを出力する場合は変位に対応したプログラムカウンタ更新パターンを出力するプログラムカウンタ値生成手段，事前に決められた規則に従い、前記リソースアクセスフラグ生成手段の出力するビットパターンと前記メモリアクセスフラグ生成手段の出力するビットパターンと前記プログラムカウンタ値生成手段の出力するプログラムカウンタ更新パターンとから前記リソースビットパターンを生成するとともに、前記リソース保持手段の保持するリソースの値と、前記メモリ値生成手段の出力するメモリの値と、前記プログラムカウンタ値生成手段がプログラムカウンタ全体の値を出力する場合は、この出力されたプログラムカウンタ全体の値とを含んだトレースデータを作成すると同時にトレースデータ出力完了指示を出力するトレースデータ生成手

段，ならびにこのトレースデータ生成手段の出力するトレースデータ出力完了指示により前記リソース保持手段の保持するリソースの値で前記前値保持手段の保持するリソースの値を更新する前値更新手段として機能させるためのプログラムを記録する。

【００４２】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照しながら詳細に説明する。

【００４３】図１は、本発明の第１の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。本実施の形態に係るトレースデータ削減方式は、メモリ１と、プロセッサ２と、リソース保持手段３と、前値保持手段４と、リソースアクセスフラグ生成手段５と、メモリアクセスフラグ生成手段６と、リソース値生成手段７と、メモリ値生成手段８と、プログラムカウンタ値生成手段９と、トレースデータ生成手段１０と、前値更新手段１１とから構成されている。

【００４４】メモリ１は、プログラムやデータを保持する。

【００４５】プロセッサ２は、メモリ１からプログラムやデータを受け取り処理を行う。また、プロセッサ２は、デバグ割り込み指示に応じて、１命令毎に自分の持つ命令語，プログラムカウンタ，レジスタなどのリソースの値を出力する。

【００４６】リソース保持手段３は、プロセッサ２から出力されたリソースの値により保持しているリソースの値を更新するとともに、リソース更新指示を出力する。

【００４７】前値保持手段４は、１命令前のデバグ割り込み発生時にリソース保持手段３が保持していたリソースの値を保持している。

【００４８】リソースアクセスフラグ生成手段５は、リソース保持手段３からのリソース更新指示を受け取ると、リソース保持手段３の保持するリソースの値と前値保持手段４の保持するリソースの値とを比較し、リソース保持手段３の保持するリソースの値と前値保持手段４の保持するリソースの値とが一致する場合には０、リソース保持手段３の保持するリソースの値と前値保持部４の保持するリソースの値とが一致しない場合には１となる、各リソースと１対１に対応したビットパターンを出力する。

【００４９】メモリアクセスフラグ生成手段６は、リソース保持手段３からのリソース更新指示を受け取ると、リソース保持手段３の保持する命令語の値からメモリ１に対してアクセスが発生したか否かを判定すると同時に、メモリ１へのアクセスがあった場合には１、メモリ１へのアクセスがなかった場合には０となる命令語のアドレス指定フィールドに対応したビットパターンを出力する。

【００５０】リソース値生成手段７は、リソースアクセスフラグ生成手段５の出力するビットパターンに基づいてリソース保持手段３の保持するリソースの値からアクセスされたリソースの値のみを選択して出力する。

【００５１】メモリ値生成手段８は、メモリアクセスフラグ生成手段６の出力するビットパターンとリソース保持手段３の保持する命令語やレジスタなどのリソースの値とから、アクセスされたメモリ１のアドレスを計算するとともに、メモリ１からアクセスされたアドレスの内容を読み出して出力する。

【００５２】プログラムカウンタ値生成手段９は、リソース保持手段３からリソースアクセス指示を受け取ると、リソース保持手段３の保持するプログラムカウンタの値と前値保持手段４の保持するプログラムカウンタの値との差分から事前に指定された変位で更新されたか否かを判断し、指定された変位で更新された場合は、その変位に対応したビットパターンを出力し、指定された変位以外で更新された場合は、対応する変位を持たないビットパターンを出力するとともに、リソース保持手段３の保持するプログラムカウンタの値を出力する。

【００５３】トレースデータ生成手段１０は、リソースアクセスフラグ生成手段５の出力するビットパターンと、メモリアクセスフラグ生成手段６の出力するビットパターンと、プログラムカウンタ値生成手段９の出力する変位のビットパターンとを事前に設定された規則に従って再構成して出力するとともに、この出力に続き、リソース値生成手段７の出力するリソースの値と、メモリ値生成手段８の出力するリソースの値とを出力する。また、プログラムカウンタ値生成手段９の出力する変位のビットパターンが対応する変位の値を持たないパターンの場合は、トレースデータ生成手段１０は、プログラムカウンタ値生成手段９の出力するプログラムカウンタの値を事前に指定された規則に従って出力する。最後に、トレースデータ生成手段１０は、すべてのトレースデータの出力を完了すると、トレースデータ出力完了指示を出力する。

【００５４】前値更新手段１１は、次の命令のトレースデータを採取する準備として、トレースデータ生成手段１０から出力されるトレースデータ出力完了指示を受け取ると、リソース保持手段３の保持するリソースの値を読み出し、読み出したリソースの値で前値保持手段４の保持するリソースの値を更新する。

【００５５】次に、このように構成された第１の実施の形態に係るトレースデータ削減方式の動作について説明する。

【００５６】まず、情報処理装置の前提条件については、従来例でも使用した図２の条件を使用する。

【００５７】次に、リソースに対するアクセスがあったか否かを示すビットとプログラムカウンタの変位とからなるビットパターンの形式を、図３のビットパターンフォーマットとする。図３のビットパターンフォーマットにおいて、ビットＲ００～Ｒ１５は、それぞれ１６本あ

るレジスタ０～１５と１対１で対応する。ビットＩＲは、命令語に対応する。ビットＡＳ１およびＡＳ２は、命令語の持つ２つのアドレス指定フィールドと１対１で対応する。ビットＩＬ２～ＩＬ０はプログラムカウンタの変位を示すビットであり、プログラムカウンタの変位の１，２または４の値がそのまま２進値として設定されるものとする。また、ビットＩＬ２～ＩＬ０の値がすべて１のとき、トレースデータ上にプログラムカウンタ全体の値が登録されていることを示すこととする。ここで、図４は、図３で定義したビットパターンに対する設定の例であり、レジスタ２，８，命令語および命令語のアドレス指定フィールド１で指定されたアドレスがアクセスされ、プログラムカウンタは変位２で更新された場合を示したものである。

【００５８】トレースデータへの出力順は、はじめに図３のビットパターンフォーマットにしたがって作成されたビットパターンが出力され、次にリソースの値が図３のビットパターンフォーマットの定義の左から順番に出力されるものとする。たとえば、レジスタ１および２の値が出力される場合は、ビットパターン，レジスタ１およびレジスタ２の内容の順でトレースデータに出力される。

【００５９】プロセッサ２は、メモリ１にあるプログラムやデータに従いプログラムを実行するとともに、デバグ割り込みが指示された場合、指定されたデバグ割り込み条件に従って、デバグ割り込みを発生させ、デバグ割り込み指示を出力すると同時に、そのときのプロセッサ２の持つプログラムカウンタや１６本のレジスタ０～１５などのリソースの値をリソース保持手段３に出力する。

【００６０】リソース保持手段３は、プロセッサ２からデバグ割り込みによるリソースの値を受け取ると、事前に指定された領域に各リソースの値を格納する。また、このとき、前値保持手段４には、１命令前にリソース保持手段３に保持されていたリソースの値が保持されている。リソース保持手段３は、プロセッサ２からデバグ割り込み信号を受け取ると、まず、リソースアクセスフラグ生成手段５に対し、図３に示したビットパターンフォーマットのうち、メモリ１のアクセスがあったか否かを示すビットＡＳ１およびＡＳ２ならびにプログラムカウンタの変位を示すビットＩＬ０～ＩＬ２を除くビットＲ００～Ｒ１５およびＩＲからなるビットパターンの生成を指示する。

【００６１】リソースアクセスフラグ生成手段５は、リソース保持手段３からのビットパターン生成指示を受け取ると、リソース保持手段３の保持する命令語および１６本のレジスタ０～１５の値と前値保持手段４の保持する命令語および１６本のレジスタ０～１５の値とを比較し、値に変化のあったリソースを調べる。そして、値に変化のあったリソースに対応するビットを１、値に変化のなかったリソースに対応するビットを０としたビットパターンを生成する。たとえば、図５の１命令目の場合、レジスタ２，８および命令語がアクセスされていることから、図３に示すビットパターンフォーマットのうち、ビットＲ０２，Ｒ０８およびＩＲが１、ビットＲ００，Ｒ０１，Ｒ０３，Ｒ０７，Ｒ０９およびＲ１５が０となるビットパターンを生成し、これをリソース値生成手段７に出力する。

【００６２】これと平行して、リソース保持手段３は、プロセッサ２からデバグ割り込み指示を受け取ると、メモリアクセスフラグ生成手段６に対してメモリ１に対するアクセスがあったか否かを示すビットＡＳ１およびＡＳ２に対応するビットパターンの生成を指示する。

【００６３】メモリアクセスフラグ生成手段６は、リソース保持手段３からビットパターン生成指示を受け取ると、リソース保持手段３の保持する命令語の値を調べ、メモリアクセスがあったか否か、またメモリアクセスがあった場合には命令語のメモリ指定フィールドのいずれで指定されたアドレスかを調べる。そして、メモリ１に対するアクセスがあった場合は、メモリアクセスのあった領域を指定しているメモリ指定フィールドに対応するビットを１、メモリアクセスがなかったメモリ指定フィールドに対応するビットを０とした、図３のビットパターンフォーマットの中のビットＡＳ１およびＡＳ２に対応したビットパターンを生成し、これをメモリ値生成手段８に出力する。たとえば、図５の１命令目の場合、アドレス指定フィールド１で指定されたアドレスにアクセスがあることから、図３に示すビットパターンフォーマットのうち、ビットＡＳ１が１、ビットＡＳ２が０となるビットパターンをメモリ値生成手段８に出力する。

【００６４】同時に、リソース保持手段３は、プロセッサ２からデバグ割り込み指示を受け取ると、プログラムカウンタ値生成手段９に対してプログラムカウンタの変位を示すビットＩＬ２～ＩＬ０に対応したビットパターンの生成を指示する。

【００６５】プログラムカウンタ値生成手段９は、リソース保持手段３からのビットパターン生成指示を受け取ると、リソース保持手段３の保持するプログラムカウンタの値と前値保持手段４の保持する１命令前のプログラムカウンタの値とを比較し、その変位に対応するビットパターンを生成し、トレースデータ生成手段１０に出力する。また、プログラムカウンタ値生成手段９は、求めた変位に対応するビットパターンがない場合は、どの変位とも対応しないビットパターンとともに、プログラムカウンタ全体の値も併せて、トレースデータ生成手段１０に出力する。たとえば、図５の１命令目の場合、命令語長の２でプログラムカウンタが更新されるため、図３に示すビットパターンフォーマットのビットＩＬ２～ＩＬ０には、２進値の２に対応する０，１および０となるビットパターンが生成され、これがトレースデータ生成

手段１０に出力される。また、図５の５命令目の場合、プロラムカウンタの値が１０１４から２０００に更新されているため、事前に規定された変位の１，２または４のどれとも対応がとれないことから、ビットＩＬ２～ＩＬ０がすべて１となるビットパターンがトレースデータ生成手段１０に出力されるとともに、プログラムカウンタの値である２０００もトレースデータ生成手段１０に出力される。

【００６６】リソース値生成手段７は、リソースアクセスフラグ生成手段５からビットパターンを受け取ると、このビットパターンのうちの１となっているビットに対応したリソースの値をリソース保持手段３から取り出し、トレースデータ生成手段１０に出力する。たとえば、図５の１命令目の場合、先に説明したようにリソースアクセスフラグ生成手段５から、図４のＲ０２，Ｒ０８，ＩＲに対応したビットが１となるビットパターンが出力されるため、リソース値生成手段７は、リソース保持手段３からレジスタ２，８および命令語の値である４４４，６６６および１０を取り出し、トレースデータ生成手段１０に出力する。

【００６７】メモリ値生成手段８は、メモリアクセスフラグ生成手段６の出力するビットパターンを受け取ると、このビットパターンのうちの１となっているアドレス指定フィールドを調べる。もし、１に設定されているアドレス指定フィールドがある場合、まずリソース保持手段３から命令語を読み出し、対応するアドレス指定フィールドの内容を取り出す。続いて、メモリ値生成手段８は、アドレス指定フィールドの内容をリソース保持手段３の保持するリソースの値を使って展開し、アクセスされたメモリ１のアドレスを求める。そして、求められたアドレスに対応する内容をメモリ１から取り出し、この内容をトレースデータ生成手段１０に出力する。たとえば、図５の１命令目の場合、先に示したようにビットＡＳ１に対応したビットを１に設定したビットパターンがメモリアクセスフラグ生成手段６から出力されるため、メモリ値生成手段８は、リソース保持手段３の保持する命令語のアドレス指定フィールド１の部分を取り出し、アクセスされたアドレスを計算する。図５の１命令目の場合、ここで計算されたアドレスが１２０となるため、メモリ値生成手段８は、アドレスの１２０を指定して、メモリ１から８８８の値を取り出し、これをトレースデータ生成手段１０に出力する。

【００６８】トレースデータ生成手段１０は、リソースアクセスフラグ生成手段５，メモリアクセスフラグ生成手段６およびプログラムカウンタ値生成手段９の出力する各ビットパターンを受け取ると、図３のビットパターンフォーマットに従ったビットパターンを構築し、トレースデータとして出力する。次に、トレースデータ生成手段１０は、アクセスのあったリソースやメモリがある場合、リソース値生成手段７の出力するリソースの値およびメモリ値生成手段８の出力するリソースの値を図３のビットパターンフォーマットの左に指定されているものから順にトレースデータとして出力する。また、トレースデータ生成手段１０は、プログラムカウンタ値生成手段９から出力されたビットＩＬ２～ＩＬ０に対応するビットパターンがすべて１の場合、プログラムカウンタ値生成手段９から出力されるプログラムカウンタの値もトレースデータとして出力する。

【００６９】したがって、最終的にトレースデータ生成手段１０から出力されるトレースデータは、たとえば、図５の１命令目の場合、図３のビットパターンフォーマットにおけるビットＲ０２，Ｒ０８，ＡＳ１およびＩＲが１、ビットＩＬ２～ＩＬ０が０，１および０であることから、まずはじめに、”００１０００００１０００００００１１００１０”のビットパターンが出力される。次に、図３のビットパターンフォーマットの左から定義されている順番にしたがい、レジスタ２，レジスタ８，命令語およびアドレス指定フィールド１で指定されたアドレスにあるメモリの値の順に４４４，６６６，１０および８８８がトレースデータとして出力される。

【００７０】つまり、図５の１命令目の場合、以下のようなトレースデータが生成されることになる。

【００７１】

００１０００００１０００００００１１００１０，
４４４，
６６６，
１０，
８８８

【００７２】また、図５の５命令目の場合、アクセスされたリソースは命令語のみである。一方、プログラムカウンタについては１命令前との変位が、２０００－１０１４＝９８６であり、事前にビットパターンと対応づけられた１，２または４の値でないため、プログラムカウンタ値生成手段９は、ビットＩＬ２～ＩＬ０を１，１および１と設定したビットパターンをトレースデータ生成手段１０に出力する。

【００７３】この場合、トレースデータ生成手段１０は、命令語以外にプログラムカウンタ値生成手段７から出力されたプログラムカウンタの値も、トレースデータの最後に出力する必要がある。つまり、図５の５命令目の場合、トレースデータ生成手段１０は、命令語に対応するビットＩＲを１、プログラムカウンタの変位に対応するビットＩＬ２～ＩＬ０を１，１および１とした、”００００００００００００００１００１１１”のビットパターンをトレースデータとして出力した後、リソース値生成手段７から出力される命令語の２０を出力し、その後、プログラムカウンタ値生成手段９から出力されるプログラムカウンタの値の２０００を出力する必要がある。

【００７４】したがって、図５の５命令目の場合、以下

のようなトレースデータが生成されることになる。

【００７５】

０００００００００００００００１００１１１，

２０，

２０００

【００７６】また、トレースデータ生成手段１０は、１命令分のトレースデータをすべて出力し終えると、トレースデータ出力完了指示を、前値更新手段１１に出力する。

【００７７】前値更新手段１１は、トレースデータ出力手段１０からのトレースデータ出力完了指示を受け取ると、リソース保持手段３の保持するリソースの値を読み出し、前値保持手段４の保持するリソースの値を更新する。これにより、前値保持手段４は、次の命令に対して１命令前のリソースの値を保持することになる。

【００７８】以上のようにして図５の５つの命令で生成されるトレースデータを考えると、各命令毎に出力されるトレースデータは図６のようになる。

【００７９】図６において、トレースデータの列が出力されるトレースデータに当たる。情報の内容の列は、トレースデータの値がどのリソースの値を示している情報であり、トレースデータに含まれる情報ではない。

【００８０】ここで、第１の実施の形態に係るトレースデータ削減方式におけるトレースデータの情報量について考える。

【００８１】まず、各リソースがアクセスされたプログラムカウンタの変位を記憶するビットパターンの情報量を、図３のビットパターンフォーマットから２２ビットとする。また、それ以外の情報については、従来例と比較するため、従来例でも使用した図７のトレースデータ上の情報量を使う。

【００８２】この条件により求めたビット単位の情報量が図６中の情報量の列であり、各欄の値が、個々のビットパターンやリソースのサイズを示している。そして、合計の行に示された値が、５命令分のトレースデータ全体のビット数である。

【００８３】図６の情報量の合計から、本実施の形態での５命令分のトレースデータ量は、４６２ビットとなっており、図１５の従来例の６２７ビットに比べ、トレースデータの情報量を削減できていることになる。

【００８４】次に、本発明の第２の実施の形態について図面を参照して詳細に説明する。

【００８５】図８を参照すると、本発明の第２の実施の形態に係るトレースデータ削減方式は、キーボード等の入力装置８１と、メモリ１およびプロセッサ２を含む情報処理装置８２と、記憶装置８３と、ディスプレイ等の出力装置８４と、ソフトウェアであるトレースプログラム８６を記録した記録媒体８５とから構成されている。記録媒体８５は、磁気ディスク，半導体メモリ，その他の記録媒体であってもよい。

【００８６】図９，図１０および図１１は、第２の実施の形態に係るトレースデータ削減方式をトレースプログラム８６で実現した場合の処理を示すフローチャートである。

【００８７】トレースプログラム８６は、記録媒体８５から情報処理装置８２を介して記憶装置８３に読み込まれ、情報処理装置８２の動作を制御する。情報処理装置８２は、トレースプログラム８６により、以下の処理を実行する。

【００８８】デバグ割り込みが発生すると、プロセッサ２は、自分の持つリソースの値を出力する。

【００８９】リソース保持手段３は、プロセッサ２から出力されたリソースの値を、リソース保持手段３の持つリソースの値を格納する配列ＣＲＢに格納する。

【００９０】次に、リソースアクセスフラグ生成手段５は、リソースアクセスフラグ格納用変数ＲＡＦおよびワーク用変数Ｎをそれぞれ０に初期化し（ステップＳ１０１）、配列要素ＣＲＢ（Ｎ）と前値保持手段４の持つリソースの値を格納する配列要素ＢＲＢ（Ｎ）とが一致するかどうかを判定する（ステップＳ１０２）。両者が一致していれば、リソースアクセスフラグ生成手段５は、リソースアクセスフラグ格納用変数ＲＡＦの０ビット目から１６ビット目までをリソースアクセスフラグ格納用変数ＲＡＦの０ビット目から１５ビット目と０とをビット連結したものとする（ステップＳ１０３）。両者が一致していなければ、リソースアクセスフラグ生成手段５は、リソースアクセスフラグ格納用変数ＲＡＦの０ビット目から１６ビット目までをリソースアクセスフラグ格納用変数ＲＡＦの０ビット目から１５ビット目と１とをビット連結したものとする（ステップＳ１０４）。次に、リソースアクセスフラグ生成手段５は、ワーク用変数Ｎを１つインクリメントし（ステップＳ１０５）、ワーク用変数Ｎが１６未満であれば（ステップＳ１０６でノー）、ステップＳ１０２に制御を戻してステップＳ１０２～ステップＳ１０６を繰り返し、ワーク用変数Ｎが１６以上であれば（ステップＳ１０６でイエス）、処理を終了する。

【００９１】続いて、メモリアクセスフラグ生成手段６は、メモリアクセスフラグ格納用変数ＭＡＦおよびワーク用変数Ｎをそれぞれ０に初期化し（ステップＳ２０１）、配列ＣＲＢ内のプログラムカウンタのＮ番目のアドレスシラブルのメモリアクセスを確認する（ステップＳ２０２）。メモリアクセスがなければ、メモリアクセスフラグ生成手段６は、メモリアクセスフラグ格納用変数ＭＡＦの０ビット目から２ビット目までをメモリアクセスフラグ格納用変数ＭＡＦの１ビット目と０とをビット連結したものとし（ステップＳ２０３）、メモリアクセスがあれば、メモリアクセスフラグ格納用変数ＭＡＦの０ビット目から２ビット目までをメモリアクセスフラグ格納用変数ＭＡＦの１ビット目と１とをビット連結し

たものとする（ステップＳ２０４）。次に、メモリアクセスフラグ生成手段６は、ワーク用変数Ｎを１つインクリメントし（ステップＳ２０５）、ワーク用変数Ｎが２未満であれば（ステップＳ２０６でノー）、ステップＳ２０２に制御を戻してステップＳ２０２～ステップＳ２０６を繰り返し、ワーク用変数Ｎが２以上であれば（ステップＳ２０６でイエス）、処理を終了する。

【００９２】次に、リソース値生成手段７は、リソース数格納用変数ＲＳＮを０に、テンポラリ変数ＴＲＡＦをリソースアクセスフラグ格納用変数ＲＡＦに、ワーク用変数Ｎを０にそれぞれ初期化し（ステップＳ３０１）、テンポラリ変数ＴＲＡＦの０ビット目から１ビット目までが０であるかどうかを判定する（ステップＳ３０２）。０でなければ、リソース値生成手段７は、リソースの値を保持する配列要素ＲＶＴ（ＲＳＮ）に配列要素ＣＲＢ（Ｎ）を代入し、ソース数格納用変数ＲＳＮを１つインクリメントする（ステップＳ３０３）。次に、リソース値生成手段７は、テンポラリ変数ＴＲＡＦの０ビット目から１６ビット目をテンポラリ変数ＴＲＡＦの０ビット目から１５ビット目と０とをビット結合したものとし、ワーク用変数Ｎを１つインクリメントする（ステップＳ３０４）。次に、リソース値生成手段７は、ワーク用変数Ｎが１６未満であれば（ステップＳ３０５でノー）、ステップＳ３０２に制御を戻してステップＳ３０２～ステップＳ３０５を繰り返し、ワーク用変数Ｎが１６以上であれば（ステップＳ３０５でイエス）、処理を終了する。

【００９３】続いて、メモリ値生成手段８は、アドレスシラブル数格納用変数ＡＳＮを０に、テンポラリ変数ＴＭＡＦをメモリアクセスフラグ格納用変数ＭＡＦに、ワーク用変数Ｎを０にそれぞれ初期化し（ステップＳ４０１）、テンポラリ変数ＴＭＡＦの０ビット目から１ビット目までが０であるかどうかを判定する（ステップＳ４０２）。０でなければ、メモリ値生成手段８は、ＡＳＶ＝ＣＲＢに含まれる命令語のＮの値に対応したアドレスシラブルを展開したアドレス値とし（ステップＳ４０３）、アクセスされたメモリの値を保持する配列要素ＭＶＴ（ＡＳＮ）にメモリ（ＡＳＶ）の値を代入し、アドレスシラブル数ＡＳＮを１つインクリメントする（ステップＳ４０４）。次に、メモリ値生成手段８は、テンポラリ変数ＴＭＡＦの０ビット目から２ビット目を、テンポラリ変数ＴＭＡＦの１ビット目と０とをビット結合したものとし、ワーク用変数Ｎを１つインクリメントする（ステップＳ４０５）。続いて、メモリ値生成手段８は、ワーク用変数Ｎが２未満であれば（ステップＳ４０６でノー）、ステップＳ４０２に制御を戻してステップＳ４０２～ステップＳ４０６を繰り返し、ワーク用変数Ｎが２以上であれば（ステップＳ４０６でイエス）、処理を終了する。

【００９４】次に、プログラムカウンタ値生成手段９は、プログラムカウンタ差分格納用変数ＰＣＤを配列ＣＲＢ内のプログラムカウンタの値からリソース値格納用配列ＢＲＢ内のプログラムカウンタの値を引いたものにし（ステップＳ５０１）、プログラムカウンタの差分ＰＣＤが１，２または４であるかどうかを判定する（ステップＳ５０２）。そうでなければ、プログラムカウンタ値生成手段９は、ビットパターン格納用変数ＩＬを７とし、プログラムカウンタ値保持変数ＰＣＴをリソース値格納配列ＣＲＢ内のプログラムカウンタの値とし（ステップＳ５０３）、そうであれば、ビットパターン格納用変数ＩＬをプログラムカウンタ差分格納用変数ＰＣＤとし（ステップＳ５０４）、処理を終了する。

【００９５】続いて、トレースデータ生成手段１０は、ビットパターンフィールド値格納用変数ＢＰＦをリソースアクセスフラグ格納用変数ＲＡＦ，メモリアクセスフラグ格納用変数ＭＡＦおよびビットパターン格納用変数ＬＮのビット結合とし（ステップＳ６０１）、ビットパターンフィールド値格納用変数ＢＰＦをトレースデータとして出力し（ステップＳ６０２）、ワーク用変数Ｎを０とする（ステップＳ６０３）。次に、トレースデータ生成手段１０は、ワーク用変数Ｎがリソース数格納用変数ＲＳＮ以上であるかどうかを判定し（ステップＳ６０４）、ワーク用変数Ｎがリソース数格納用変数ＲＳＮ未満であれば、リソースの値を保持する配列要素ＲＶＴ（Ｎ）をトレースデータとして出力し（ステップＳ６０５）、ワーク用変数Ｎを１つインクリメントして（ステップＳ６０６）、ステップＳ６０４に制御を戻す。ステップＳ６０４で、ワーク用変数Ｎがリソース数格納用変数ＲＳＮ以上であれば、トレースデータ生成手段１０は、ワーク用変数Ｎを０とし（ステップＳ６０７）、ワーク用変数Ｎがアドレスシラブル数格納用変数ＡＳＮ以上であるかどうかを判定する（ステップＳ６０８）。ワーク用変数Ｎがアドレスシラブル数格納用変数ＡＳＮ未満であれば、メモリの値を保持する配列要素ＭＶＴ（Ｎ）をトレースデータとして出力し（ステップＳ６０９）、ワーク用変数Ｎを１つインクリメントして（ステップＳ６１０）、ステップＳ６０８に制御を戻す。ステップＳ６０８で、ワーク用変数Ｎがアドレスシラブル格納用変数ＡＳＮ以上であれば、トレースデータ生成手段１０は、ビットパターン格納用変数ＩＬが７かどうかを判定し（ステップＳ６１１）、そうであれば、プログラムカウンタ値保持変数ＰＣＴをトレースデータとして出力して（ステップＳ６１２）、処理を終了する。

【００９６】次に、前値更新手段１１は、リソース保持手段３の配列ＣＲＢを読み出し、前値保持手段４に出力する。

【００９７】最後に、前値保持手段４は、前値更新手段１１の出力にて配列ＢＦＢを更新する。

【００９８】

【発明の効果】第１の効果は、トレースデータのデータ

量を削減できることにある。これにより、従来採取が不可能であった大規模なプログラムのトレースデータが採取可能となり、大規模なプログラムの性能評価を行うことを可能とすることである。その理由は、トレースデータとして採取すべき情報処理装置の持つ情報には、いくつかの特性があり、その特性に着目することにより、採取すべきトレースデータの量を減らすことができるからである。

【００９９】たとえば、通常の情報処理装置では、１命令毎にプログラムカウンタ，命令語，レジスタなどの複数のリソースがアクセスされる。一方、トレースデータへの出力が必要となるレジスタなどのすべてのリソースの数は、数十程度の場合が一般である。その場合、リソース毎に識別情報を与えるより、本発明のように各リソースと１対１で対応したビットパターンで表現した方が、同時に複数のリソースに対するアクセスの有無を表現できることから、全体としての情報量を減らすことができる。

【０１００】また、プログラムカウンタは、多くの場合、プログラムカウンタ全体に対し、比較的小さな数種類の値でインクリメントされるだけである。したがって、プログラムカウンタ全体を１命令毎にトレースデータとして記録することは無駄であり、本発明のように通常は変位に対応した少ないビット数をトレースデータに記録し、このビット幅で表現できなくなったときのみ、プログラムカウンタ全体をトレースデータに記録することにより、トレースデータの情報量を減らすことができる。

【０１０１】さらに、ある命令がどのアドレスを参照したかは、命令語とアドレスを指定するために使用されるレジスタの値とが既知であれば、計算により求めることが可能であり、トレースデータとして、アクセスされたメモリのアドレスを記録することも無駄である。つまり、本発明のようにメモリをアクセスした命令語内のアドレス指定フィールドを識別する手段のみを提供すれば、多くのビット数を必要とするアドレスの値をトレースデータ内に保持する必要はなく、トレースデータの情報量を減らすことができる。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。

【図２】第１の実施の形態を説明するための情報処理装置の条件を示す図である。

【図３】第１の実施の形態で使用するビットパターンのフォーマットを示す図である。

【図４】図３のビットパターンフォーマットでのビットパターン例を示す図である。

【図５】第１の実施の形態で使用する命令実行イメージを示す図である。

【図６】図５の命令実行イメージを第１の実施の形態を使って採取したトレースデータ列を示した図である。

【図７】第１の実施の形態で採取されるトレースデータ上の各リソースの情報量を示す図である。

【図８】本発明の第２の実施の形態に係るトレースデータ削減方式の構成を示すブロック図である。

【図９】第２の実施の形態に係るトレースデータ削減方式の処理の前部を示すフローチャートである。

【図１０】第２の実施の形態に係るトレースデータ削減方式の処理の中程部を示すフローチャートである。

【図１１】第２の実施の形態に係るトレースデータ削減方式の処理の後部を示すフローチャートである。

【図１２】従来例のおける１つのリソースに対する単位情報の例を示す図である。

【図１３】図１２の単位情報に対する設定の例を示す図である。

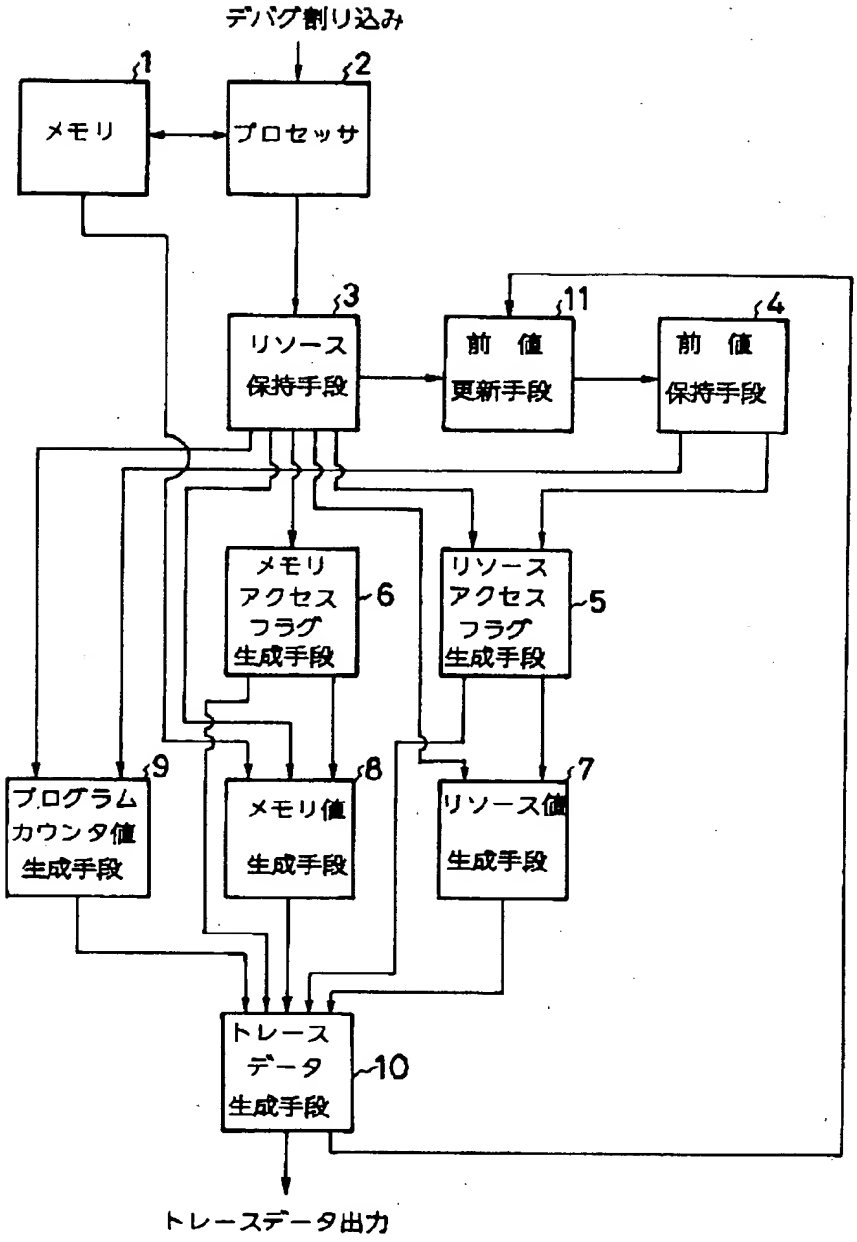
【図１４】従来例で使用するリソースと識別情報との対応表である。

【図１５】従来例で図５の命令実行イメージを使って採取したトレースデータイメージを示す図である。

【符号の説明】

１　メモリ
２　プロセッサ
３　リソース保持手段
４　前値保持手段
５　リソースアクセスフラグ生成手段
６　メモリアクセスフラグ生成手段
７　リソース値生成手段
８　メモリ値生成手段
９　プログラムカウンタ値生成手段
１０　トレースデータ生成手段
１１　前値更新手段
２１　識別情報
２２　位置情報
２３　内容
８１　入力装置
８２　情報処理装置
８３　記憶装置
８４　出力装置
８５　記録媒体
８６　トレースプログラム

【図１】



【図２】

情報処理装置条件

レジスタ数	１６
アドレス指定フィールド数	２
プログラムカウンタ初期値	１０００

【図７】

トレースデータ上の情報量

情報量		ビット数
識別情報		3
位置情報	レジスタ	4
	メモリ	３２
内容	命令語	命令長
	プログラムカウンタ	３２
	レジスタ	３２
	メモリ	３２

【図１４】

識別情報対応表

命令語	I
プログラムカウンタ	P
レジスタ	R
メモリ	M
区切り情報	S

【図３】

ビットパターンフォーマット

R00	R01	R02	R03	R04	R05	R06	R07	R08	R09	R10	R11	R12	R13	R14	R15	IR	AS1	AS2	IL2	IL1	IL0

【図４】

ビットパターン例

0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0

【図１２】

単位情報

識別情報	位置情報	内容
21	22	23

【図１３】

単位情報の例

R	1	１２３４
21	22	23

【図５】

命令実行イメージ

1命令目			
リソース	位　置	フィールド位置	アクセス内容
命令語	－	－	１０
命令語長	－	－	１６
プログラムカウンタ	－	－	１００２
レジスタ	２	－	４４４
レジスタ	８	－	６６６
メモリ	１２０	１	８８８

2命令目			
リソース	位　置	フィールド位置	アクセス内容
命令語	－	－	４５６
命令語長	－	－	３２
プログラムカウンタ	－	－	１００６
レジスタ	１５	－	１３５
メモリ	４８０	２	１０１０

3命令目			
リソース	位　置	フィールド位置	アクセス内容
命令語	－	－	７８９
命令語長	－	－	３２
プログラムカウンタ	－	－	１０１０
レジスタ	７	－	１２４

4命令目			
リソース	位　置	フィールド位置	アクセス内容
命令語	－	－	７８９
命令語長	－	－	３２
プログラムカウンタ	－	－	１０１４
レジスタ	７	－	２４８

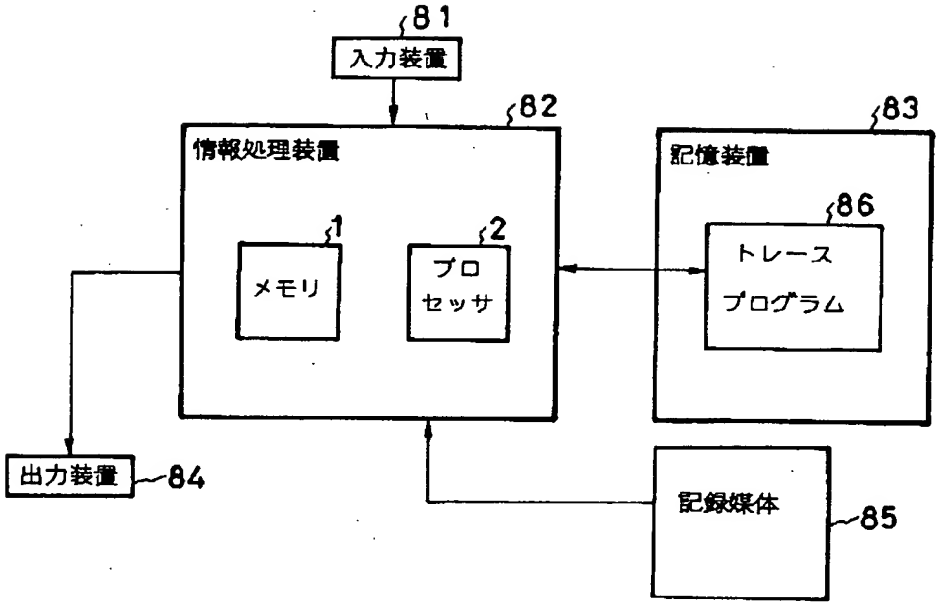
5命令目			
リソース	位　置	フィールド位置	アクセス内容
命令語	－	－	２０
命令語長	－	－	１６
プログラムカウンタ	－	－	２０００

【図６】

トレースデータイメージ

		トレースデータ	情報の内容	情報量　(bit)
１命令目	ビットパターン	00100000100000000100000		22
	データ	４４４	レジスタ２	32
	データ	６６６	レジスタ８	32
	データ	８８８	メモリ（フィールド１）	32
	データ	１０	命令語	16
２命令目	ビットパターン	00000000000000001101100		22
	データ	１３５	レジスタ１５	32
	データ	８８８	メモリ（フィールド２）	32
	データ	４５６	命令語	32
３命令目	ビットパターン	00000001000000000100010		22
	データ	１２４	レジスタ７	32
	データ	７８９	命令語	32
４命令目	ビットパターン	00000001000000000000010		22
	データ	４４４	レジスタ７	32
５命令目	ビットパターン	00000000000000000100111		22
	データ	２０	命令語	16
	データ	２０００	プログラムカウンタ	32
合計				462

【図８】

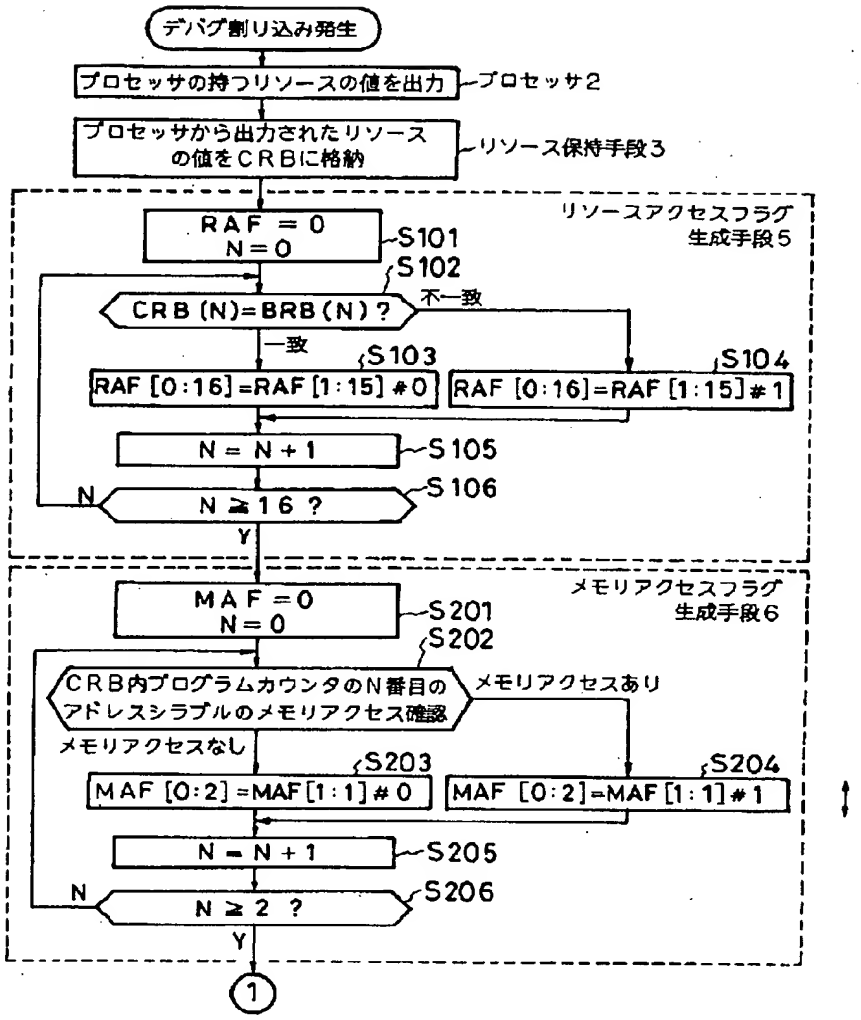


【図１５】

トレースデータイメージ

	トレースデータ			
	識別情報	位置情報	内　容	情報量（bit）
１命令目	S	－	－	3
	P	－	１００２	35
	I	－	１０	19
	R	2	４４４	39
	R	8	６６６	39
	M	１２０	８８８	67
２命令目	S	－	－	3
	P	－	１００６	35
	I	－	４５６	35
	R	１５	１３５	39
	M	４８０	１０１０	67
３命令目	S	－	－	3
	P	－	１０１０	35
	I	－	７８９	35
	R	7	１２４	39
４命令目	S	－	－	3
	P	－	１０１４	35
	R	7	２４８	39
５命令目	S	－	－	3
	P	－	２０００	35
	I	－	２０	19
合計				627

【図９】



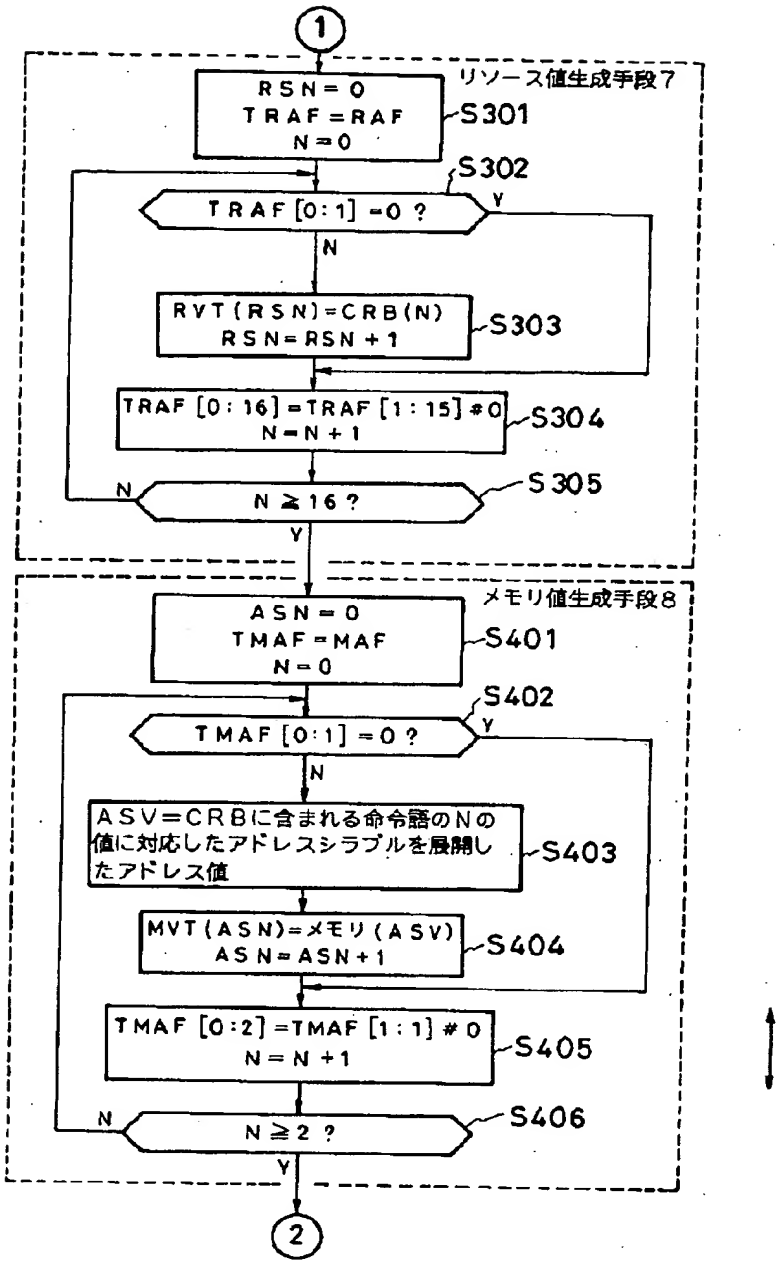
x (y)　：xの配列のy番目の位置を指す。

x [n:m]：xの変数のnビット目からmビット分を示す。nは0オリジン、pビットの変数の場合、上位ビット側を0、下位ビット側をp－1とする。

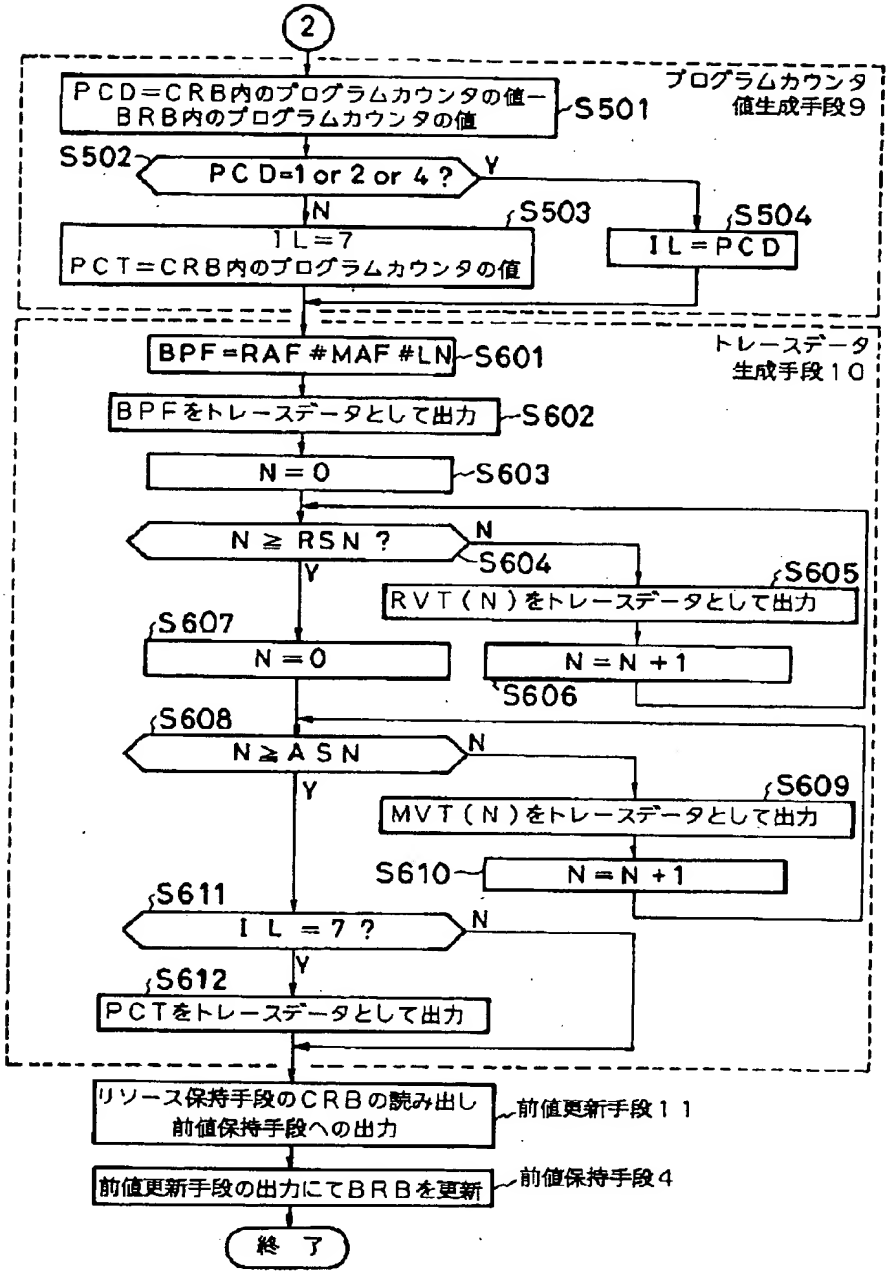
#　：ビット連結を示す。x＝y # zと記述した場合、y＝101、z＝110とすると、zは101110となる。

メモリ(x)：メモリのアドレスxを示す。

【図１０】


1
リソース値生成手段7
RSN=0
TRAF=RAF
N=0
S301
S302
TRAF[0:1]=0 ?
Y
N
RVT(RSN)=CRB(N)
RSN=RSN+1
S303
TRAF[0:16]=TRAF[1:15]#0
N=N+1
S304
N
N≧16 ?
S305
Y
メモリ値生成手段8
ASN=0
TMAF=MAF
N=0
S401
S402
TMAF[0:1]=0 ?
Y
N
ASV=CRBに含まれる命令語のNの値に対応したアドレスシラブルを展開したアドレス値
S403
MVT(ASN)=メモリ(ASV)
ASN=ASN+1
S404
TMAF[0:2]=TMAF[1:1]#0
N=N+1
S405
N
N≧2 ?
S406
Y
2

【図１１】


2
プログラムカウンタ値生成手段９
PCD＝CRB内のプログラムカウンタの値－BRB内のプログラムカウンタの値
S501
S502
PCD=1 or 2 or 4 ?
Y
N
S503
IL＝7
PCT＝CRB内のプログラムカウンタの値
S504
IL＝PCD
トレースデータ生成手段１０
BPF=RAF #MAF #LN
S601
BPFをトレースデータとして出力
S602
N＝0
S603
N ≧ RSN ?
N
S604
S605
RVT(N)をトレースデータとして出力
S607
N＝0
N＝N＋1
S606
S608
N≧ASN
N
S609
MVT(N)をトレースデータとして出力
S610
N＝N＋1
S611
IL＝7 ?
N
S612
PCTをトレースデータとして出力
リソース保持手段のCRBの読み出し前値保持手段への出力
前値更新手段１１
前値更新手段の出力にてBRBを更新
前値保持手段４
終了